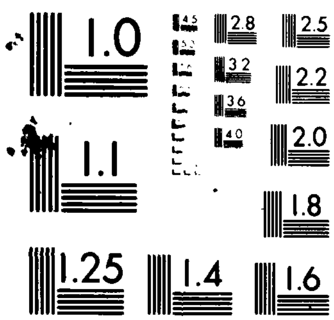
AD-A107 110 MASSACHUSETTS UNIV AMHERST DEPT OF MATHEMATICS AND S--ETC F/G 9/2
CLUSTER ANALYSIS ALGORITHMS FOR IMAGE SEGMENTATION.(U)
JUL 81 M F JANOWITZ N00014-79-C-0629
UNCLASSIFIED TR-J8102 NL

END
DATE
FILMED
12 81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Technical Report J8102

CLUSTER ANALYSIS ALGORITHMS FOR IMAGE SEGMENTATION

M. F. Janowitz
Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA 01003

July, 1981

Cluster Analysis Algorithms For Image Segmentation

M. F. Janowitz *

1. Introduction. The goal of this research is the investigation of the ability of certain cluster techniques to segment monochromatic data collected by remote sensing devices. The particular example that will be considered consists of temperature data collected by the Air Force Geophysical Laboratory at Hanscom Air Force Base. The work was done on three levels: underlying motivations, simulations, and real data analysis. In this particular paper, we shall restrict our consideration to a single feature, with multiple feature selection to be investigated at a later date. Section 2 presents the motivation for both the cluster techniques and the statistics that they use, while section 3 briefly explains the techniques. In section 4 some simulated data is analyzed, and finally in section 5, the techniques are applied to actual infrared temperature data. The paper concludes with an indication of possible future directions for the current research effort.

2. Basic Assumptions. Here is a list of the assumptions that underly all of the cluster algorithms. They are essentially due to Bryant [1], though they also bear a relation to the axioms that appear

* Research supported by ONR Contract N00014-79-C-0629 as well as by grants from the University of Massachusetts Computer Center.

in Coleman and Andrews [2], p. 775.

A1. The input data consists of an $m \times n$ matrix A, each of whose entries is a positive integer.

A2. The entries of A represent values of some function F defined on a rectangular region of a plane. This function might represent temperature, brightness, or some other attribute of the plane. Each entry of A corresponds to the average value of F in a square region of the plane, with adjacent entries corresponding to adjacent regions.

A3. There is a natural division of the plane into regions $R_1, R_2, \ldots, R_t$. Each such region is connected, has a smooth boundary, and can be characterized by the values of F. It might be that the regions R_i correspond to distinct fixed values of F, or that they might be distinguished by some sort of texture measure applied to F, or possibly some combination of features of F.

A4. The observer has no direct knowledge of the number of regions, their location, or the values of F within the given regions.

A5. The values of F are read by some sort of sensing device, but this process has some additive noise associated with it. Thus the observed data is a random variable

$$G = \mathrm{NINT}\ (F+N),$$

where N is a random noise variable whose restriction to region R_i has expected value 0 and variance σ_i^2, and NINT denoted the function that rounds a real number off to the integer to which it is closest. No

further assumptions are made about the nature of the distribution of N.

A6. Near the boundary of a region, the sensing device will sometimes be averaging values of F from more than one region. This can produce values of G that do not seem to belong to either of the regions in question. These boundary points may appear to represent a region of their own, or may even seem to be part of some other region that is not located near the boundary. One must therefore be suspicious of regions that appear to be very narrow, or are disconnected, or widely dispersed through the plane.

The cluster algorithms all operate on the following basis. Somehow, a finite collection $c_1, c_2, \ldots, c_k$ of numbers is selected. These are the <u>cluster centers.</u> A single feature of G is chosen. On the basis of that feature, each point (=pixel) is assigned to the cluster center to which it is closest. This produces k clusters $C_1, C_2, \ldots, C_k$. In view of the fact that the desired regions $R_1, R_2, \ldots, R_t$ are spatially connected with smooth boundaries (see A3), the algorithms all assume that the cluster C_i whose members have the highest <u>dispersion</u> will be the cluster that is least likely to be a subset of one of the R_j regions. It is deleted, and its members reassigned to produce $k-1$ clusters with cluster centers $c_1', c_2', \ldots, c_{k-1}'$. The process continues with termination reached when a single cluster is produced. or when the dispersion falls below some preassigned threshold.

To compute the dispersion D of cluster C_i, one looks at each member of C_i, computes the proportion of its 8 immediate neighbors that are <u>not</u> members of C_i, and averages over all members of C_i. The idea

behind using the dispersion measure is that the regions R_j are regions on which the selected feature is more or less constant, so an interior point of of R_j ought to have a low dispersion.

With a relatively low noise level, the cluster algorithms work fairly well on the observed data, but in the presence of a significant amount of noise, they tend to merge the clusters into a single region. This problem can be partially overcome by using a suitable prefilter. Those that were tried include the median of 3 by 3 neighborhoods, the mean of 3 by 3, 5 by 5, and 7 by 7 neighborhoods, and the 3 by 3 mean iterated 2 and 3 times. These last filters are denoted respectively as MM and MMM. They give unequal weight to the neighbors of a pixel, and are of some interest because their use enables the dispersion measure to effectively determine when it is close to the expected value of G within a single region R. They work quite well with both gaussian and uniform noise, and tend to provide less distortion of the original data than would a comparable s by s filter. Similar observations were made in Rosenfeld and Kak [4]. p. 161. The relative weights that the MM and MMM filters give to the neighbors of a pixel are illustrated in Figs. 1 and 2. Naturally, this idea can be extended even further. Let $M^{(k)}$ be the filter produced by k iterations of a 3 by 3 mean filter. It is then easy to show that the value of $M^{(k)}$ at a given pixel is based upon a (2k-1) by (2k-1) filter centered at that pixel. The relative weights assigned to the various points in this neighborhood satisfy the equation $a_{ij} = (a_{1i})(a_{1j})$, so all of the weights are completely determined once we know the weights in the first row. These turn out to have some interesting combinatorial interpretations. Let $M(k;r)$ denote

the value of the r+ first entry of the first row of $M^{(k)}$. It is then easy to establish the recurrence relation

(1) $M(k+1;r) = M(k;r) + M(k;r-1) + M(k;r-2).$

It is immediate from this that

(2) $(1+x+x^2)^k = \sum_{k=0}^{2k} M(k;r)x^r.$

This in turn produces the fact ([3], p. 16) that $M(k;r)$ represents the number of ways r objects may be selected from 2k objects (k kinds, 2 of a kind); thus

(3) $M(k;r) = \Sigma_t (-1)^t \binom{k}{t}\binom{k+r-3t-1}{r-3t}$

where $\binom{k}{t}$ denotes the coefficient of x^t in the expansion of $(1+x)^k$. Setting $x = 1$ in (2) produces the fact that the sum of the first row weights associated with $M^{(k)}$ is 3^k, so that the total of all of the weights is 9^k. To illustrate the operation of relation (1), we represent the following generalization of a Pascal triangle. Notice how easily each row is produced from the one immediately above it by using (1). Before

Numbers M(k;r)

k\r	0	1	2	3	4	5	6	7	8	9	10
1	1	1	1								
2	1	2	3	2	1						
3	1	3	6	7	6	3	1				
4	1	4	10	16	19	16	10	4	1		
5	1	5	15	30	45	51	45	30	15	5	1

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

Fig. 1 Relative weight given to neighbors of a pixel by the MM filter.

1	3	6	7	6	3	1
3	9	18	21	18	9	3
6	18	36	42	36	18	6
7	21	42	49	42	21	7
6	18	36	42	36	18	6
3	9	18	21	18	9	3
1	3	6	7	6	3	1

Fig. 2 Relative weight given to neighbors of a pixel by the MMM filter.

proceeding, let us see how well the dispersion serves to give the expected value of G in a single region. Selected results appear in Tables 1 through 4. Tables 1,2,3 contain data using 3 by 3, 5 by 5, and 7 by 7 filters acting on gaussian noise. It should be noted that the values of D increase as one gets further from the expected value of 0, and that the MM and MMM filters seem especially effective in locating the expected value. Table 4 contains similar results based upon noise having a uniform distribution. The next step is to present some simulations that involve the cluster algorithms, but before doing this, we shall need to describe the algorithms.

3. The cluster algorithms. Here is a brief description of the various cluster algorithms.

Cluster Method 1. Recall that the input is an m by n matrix A, each of whose entries in an integer. There are also 2 initial parameters: THRSH and CHOS. These will be explained in the description of the algorithm.

For a 30 by 30 picture, values of around 0.4, 0.25 and 0.2 seem to work pretty well for THRSH on 3 by 3, 5 by 5 and 7 by 7 filters, and CHOS is taken to be .02 × m × n.

SD	WIDTH	FILTER	LEVEL 0	1	2	3	4	5	6	7	8
2	0	Mean	.386	.566	.965	1					
		Med	.066	.312	1	1					
2	1	Mean	.028	.134	.468						
		Med	.000	.007	.199						
4	0	Mean	.629	.692	.790	.873	.830	1	1		
		Med	.432	.539	.779	.927	.807	1	1		
4	1	Mean	.233	.280	.417	.605	.784	.928			
		Med	.015	.028	.137	.396	.749	.938			
4	2	Mean	.046	.090	.209	.391	.631				
		Med	.000	.002	.023	.119	.404				
6	1	Mean	.391	.420	.480	.562	.623	.734	.812	.885	.948
		Med	.072	.093	.178	.314	.380	.637	.755	.952	1
6	2	Mean	.171	.181	.266	.379	.512	.593	.697	.774	
		Med	.008	.007	.030	.098	.264	.383	.575	.728	

Table 1. Average dispersion based on 4 trials on 30 by 30 matrix with normal distribution having mean 0 and indicated SD. Mean is a 3 by 3 mean filter and Med a 3 by 3 median filter. Level i with width k merges levels i-k,...,i,...,i+k into a single cluster before computing the dispersion.

SD	WIDTH	FILTER	LEVEL 0	1	2	3	4	5	6	7	8
2	0	Mean	.126	.608	1	1					
		MM	.003	.362	1	1					
2	1	Mean	.000	.054	.475						
		MM	.000	.001	.233						
4	0	Mean	.401	.484	.587	1	1	1	1		
		MM	.087	.189	.328	1	1	1	1		
4	1	Mean	.047	.119	.342	.644	.969	1			
		MM	.000	.020	.122	.500	1	1			
4	2	Mean	.001	.019	.126	.374	.614				
		MM	.000	.000	.011	.129	.385				
6	0	Mean	.531	.554	.644	.751	.969	1	1	1	1
		MM	.226	.279	.445	.672	1	1	1	1	1
6	1	Mean	.139	.209	.379	.553	.714	.975	1	1	1
		MM	.007	.032	.117	.349	.584	1	1	1	1
6	2	Mean	.028	.054	.161	.319	.503	.739	.979	1	
		MM	.000	.002	.026	.113	.257	.663	1	1	

Table 2. Average dispersion based on 4 trials on 30 by 30 matrix with normal distribution having mean 0 and indicated SD. Mean is a 5 by 5 mean filter and MM a 3 by 3 mean filter applied twice. See Table 1 for further explanation of symbols.

SD	WIDTH	FILTER	LEVEL 0	1	2	3	4	5	6	7	8
2	0	Mean	.050	.569	1	1					
		MMM	.001	.389	1	1					
2	1	Mean	.000	.015	.537						
		MMM	.000	.000	.329						
4	0	Mean	.243	.379	.919	1	1	1	1		
		MMM	.033	.096	1	1	1	1	1		
4	1	Mean	.004	.098	.410	.979	1	1			
		MMM	.000	.116	.179	1	1	1			
4	2	Mean	.000	.002	.057	.336	.979				
		MMM	.000	.000	.006	.116	1				
6	0	Mean	.401	.408	.515	.900	1	1	1	1	1
		MMM	.099	.104	.202	.800	1	1	1	1	1
6	1	Mean	.037	.137	.360	.734	1	1	1	1	
		MMM	.002	.020	.144	.607	1	1	1	1	1
6	2	Mean	.004	.031	.126	.324	.668	.948	1	1	
		MMM	.000	.001	.014	.089	.574	.917	1	1	

Table 3. Average dispersion based on 4 trials on 30 by 30 matrix with normal distribution having mean 0 and indicated SD. Mean is a 7 by 7 mean filter and MMM a 3 by 3 mean filter applied three times. See Table 1 for further explanation of symbols.

RANGE	SIZE	FILTER	LEVEL 0	1	2	3	4	5	6	7	8
4	3	Mean	.497	.582	.747	.975	1				
	5	Mean	.227	.469	.969	1	1				
	5	MM	.020	.204	1	1	1				
	7	Mean	.090	.475	1	1	1				
	7	MMM	.005	.254	1	1	1				
6	3	Mean	.380	.501	.682	1	1	1	1		
	5	MM	.084	.207	.425	1	1	1	1		
	7	Mean	.213	.406	1	1	1	1	1		
	7	MMM	.034	.173	1	1	1	1	1		
10	3	Mean	.562	.550	.643	.880	.925	1	1	1	1
	5	MM	.272	.283	.476	.938	.800	1	1	1	1
	7	Mean	.424	.407	.597	.953	1	1	1	1	1
	7	MMM	.120	.132	.418	1	1	1	1	1	1

Table 4. Average results of 4 trials on 30 by 30 matrix with uniform distribution having range from -RANGE to +RANGE using a SIZE by SIZE filter of the indicated type. See Table 1 for further explanation of symbols.

1. Apply an appropriate filter to A, and let B denote the resulting matrix, with each entry rounded to the nearest integer.

2. Do a frequency histogram on the values of the entries in B, omitting the outermost rows and columns of B. Choose those values whose frequency exceeds CHOS, and call these the cluster centers. Assume that $c_1, c_2, \ldots, c_t$ are these centers, arranged in ascending numerical order.

3. Assign each point in B to the cluster center to which it is closest. Let C_i be the cluster that results from the points that have been assigned to c_i.

4. Compute the dispersion of each cluster.

5. If the maximum dispersion is less than THRSH, stop here. Otherwise, choose the cluster C_i whose dispersion is maximum. Ties are broken by taking the lowest possible cluster C_i.

6. Omit the cluster center c_i from the list of centers.

7. Re-allocate the members of C_i according to the following rule: For x a member of C_i, x is assigned to the next higher or lower of the remaining cluster centers according to whether the value in B assigned to x is above or below the average of these two centers.

8. Go to step 4.

Cluster Method 2. This is identical to Method 1, except that at each iteration, the actual cluster means are computed (using the B matrix). In Step 7, x is assigned to the next higher or lower available cluster

according to which cluster mean is closest to its B value.

Cluster Method 3. This is again a modification of Step 7. When a cluster C_i is broken up, each of its members is assigned to that cluster whose mean is closest. Then all cluster means are recomputed.

Several other variants of Step 7 were tried, but they did not seem to perform as well as any of the above methods, so a detailed description of them seems unwarranted. They were based on one of the following ideas: (i) The reassignment of all members of C_i to the same cluster; (ii) The assignment of members of C_i by looking at the clusters to which the immediate neighbors of a member belong.

4. Some simulations. Each of Methods 1, 2 and 3 was applied to data with a known classification, so that their effectiveness could be compared. The underlying classification appear in Fig. 3. It is a 30 by 30 matrix with the indicated regions. The function F was constructed by assigning it values of 40, 55 and 60 in these regions. G was constructed by adding various types of gaussian and uniform noise to F. Tables 5 through 12 contain the results of the simulations. Since the format of these tables is almost identical, it should suffice to supply a few words of explanation regarding Table 5. Table 5 takes F as its input with no added noise. The values here should serve as a reference, as they show the distortion produced by the various filters, and the manner in which the dispersion can be used to determine the correct number of clusters. Thus for a 3 by 3 mean filter using Method 1, the


```
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
333333333333333333333333333333
222222222211111111111111111111
222222222221111111111111111111
222222222222111111111111111111
222222222222211111111111111111
222222222222221111111111111111
222222222222222111111111111111
222222222222222211111111111111
222222222222222221111111111111
222222222222222222111111111111
222222222222222222211111111111
222222222222222222221111111111
222222222222222222222111111111
222222222222222222222211111111
222222222222222222222221111111
222222222222222222222222111111
222222222222222222222222211111
222222222222222222222222221111
222222222222222222222222222111
222222222222222222222222222211
222222222222222222222222222221
```

FIGURE 3. SIMULATED CLUSTERS IN A 30 BY 30 REGION.

dispersion drops from .440 for 4 clusters to .088 for three clusters, and the method itself produces a correct classification of 96.9% of the points. The columns labeled Clean1 and Clean2 merit some explanation. They each represent the result of a post-cleaning operation applied to the output of the indicated cluster method. These operations are based upon two functions: OPTIMAL and CLEAN. OPTIMAL proceeds by looking at the two neighbors of each pixel immediately above and below as well as to the left and right. If either of these pairs is in the same cluster, then the point in question is assigned to that cluster. Following this, the two diagonals are examined. and a similar decision is made. CLEAN proceeds in a slightly different manner. It calls a point a boundary point if one of its four immediate neighbors (above, below or to the side) belongs to a different cluster. The cluster means are computed, and each boundary point is reassigned to the cluster to which its A value is closest. This works quite well unless the original data is noisy, and then it tends to be a dirtying rather than a cleaning operation. Clean1 operates by applying OPTIMAL CLEAN OPTIMAL CLEAN to the output, and Clean2 applies OPTIMAL OPTIMAL. Note, for example, the mixed effect of Clean1 in Tables 8 and 12, the noisiest examples that were considered. Other possibilities for cleaning include the removal of portions of clusters at points where they have width or height under 3, as well as the use of filtered data for the re-evaluation of boundary points.

To illustrate the workings of the algorithms, a step by step output

of Method 1 is presented in Fig. 4. The original classification is shown in (a). These regions are assigned values of 40, 55 and 60, and gaussian noise with a variance of 64 is added. If one has prior knowledge of the number of regions and the grey level associated with them, (b) shows the result of assigning each point to the cluster mean to which it is closest. Parts (c) through (k) illustrate the operation of Method 1, with (j) being the ouput of Method 1, and (k) the result of a cleaning operation that removes clusters with height or width 1 or 2, but leaves the outermost rows and columns unchanged. The raw output has 83% accuracy, and after cleaning this improves to 89%.

A texture measure may also of course be used as a feature. Using the regions in Fig. 3, simulated data was entered in each region so that the mean value for a region was identical,but so that the noise in the regions differed. A texture measure defined by Max $|G(i,j) - G(u,v)|$, where (u,v) ranges over the four horizontal and vertical neighbors of (i,j) was tried (see Rosenfeld and Kak [4], p. 280) with some success. The results are summarized in Table 13, What is called for of course is the joint consideration of texture with various other features. This will be done in a later work,

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.969	1	.966	.088	.440
Mean	2	.969	1	.966	.088	.440
	3	.969	1	.966	.088	.440
5 by 5	1	.905	.995	.901	.098	.347
Mean	2	.929	.998	.925	.091	.285
	3	.929	.998	.925	.091	.285
7 by 7	1	.855	.986	.851	.104	.210
Mean	2	.857	.988	.857	.104	.245
	3	.857	.988	.857	.104	.245
MM	1	.962	1	.962	.091	.421
	2	.962	1	.962	.091	.507
	3	.965	1	.962	.093	.501
MMM	1	.922	1	.917	.097	.239
	2	.926	1	.926	.096	.290
	3	.922	1	.917	.097	.284

Table 5. Fraction correct classification for data in Fig. 3 using indicated filters. See text for explanation of filters, cluster methods and cleaning operations. Left hand figure under dispersion indicates result with 3 clusters and right hand figure the result with 4 clusters.

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.907	.963	.941	.180	.406
Mean	2	.922	.957	.954	.137	.406
	3	.922	.957	.954	.137	.610
5 by 5	1	.889	.953	.887	.098	.256
Mean	2	.899	.957	.899	.077	.372
	3	.901	.948	.901	.077	.372
7 by 7	1	.866	.934	.868	.096	.239
Mean	2	.876	.942	.878	.109	.275
	3	.866	.934	.868	.096	.282
MM	1	.927	.972	.927	.098	.270
	2	.927	.972	.927	.098	.337
	3	.938	.966	.946	.096	.337
MMM	1	.924	.965	.926	.097	.217
	2	.926	.971	.922	.092	.218
	3	.938	.973	.940	.117	.272

Table 6. Fraction correct classifications for data in Fig. 3 with gaussian noise having SD 5. See caption under Table 5.

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.737	.713	.759	.345	.477
Mean	2	.772	.772	.834	.279	.518
	3	.757	.701	.815	.323	.477
5 by 5	1	.799	.806	.826	.224	.314
Mean	2	.769	.734	.804	.236	.343
	3	.800	.792	.804	.178	.377
7 by 7	1	.826	.868	.841	.176	.294
Mean	2	.826	.868	.841	.176	.214
	3	.826	.868	.841	.176	.247
MM	1	.800	.773	.833	.186	.302
	2	.785	.738	.804	.242	.299
	3	.806	.776	.839	.196	.302
MMM	1	.800	.810	.808	.149	.326
	2	.769	.738	.775	.181	.326
	3	.800	.810	.808	.149	.340

Table 7. Fraction correct classification for data in Fig. 1 with gaussian noise having SD 10. See caption under Table 5.

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.683	.602	.757	.433	.598
Mean	2	.704	.651	.737	.427	.584
	3	.463	.487	.402	.437	.433
5 by 5	1	.751	.724	.774	.221	.398
Mean	2	.632	.602	.623	.304	.322
	3	.821	.846	.813	.271	.319
7 by 7	1	.760	.723	.783	.160	.258
Mean	2	.870	.872	.876	.152	.321
	3	.868	.857	.899	.144	.321
MM	1	.774	.681	.831	.273	.480
	2	.799	.766	.823	.258	.490
	3	.767	.747	.769	.263	.375
MMM	1	.812	.744	.872	.194	.347
	2	.822	.824	.831	.171	.244
	3	.822	.824	.831	.171	.244

Table 8. Fraction correct classification for data in Fig. 1 with gaussian noise having SD 15. See caption under Table 5.

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.959	.956	.968	.100	.438
Mean	2	.956	.956	.963	.103	.425
	3	.956	.956	.963	.103	.425
5 by 5	1	.918	.963	.922	.076	.375
Mean	2	.925	.964	.925	.090	.322
	3	.905	.960	.918	.097	.431
7 by 7	1	.851	.938	.853	.110	.231
Mean	2	.868	.948	.872	.110	.280
	3	.868	.948	.872	.110	.280
MM	1	.922	.964	.918	.092	.335
	2	.938	.950	.955	.115	.335
	3	.946	.964	.943	.092	.360
MMM	1	.922	.955	.919	.096	.341
	2	.922	.955	.919	.096	.341
	3	.919	.955	.917	.097	.368

Table 9. Fraction correct classification for data in Fig. 3 with uniform noise with range from -5 to 5. See caption under Table 5.

Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.882	.919	.910	.203	.419
Mean	2	.854	.851	.901	.221	.480
	3	.886	.916	.919	.203	.419
5 by 5	1	.880	.918	.884	.125	.280
Mean	2	.880	.918	.884	.125	.253
	3	.868	.894	.885	.120	.361
7 by 7	1	.860	.909	.860	.108	.237
Mean	2	.860	.909	.860	.108	.277
	3	.860	.909	.860	.108	.277
MM	1	.891	.922	.896	.103	.292
	2	.905	.924	.920	.106	.292
	3	.903	.918	.910	.124	.326
MMM	1	.891	.932	.888	.118	.259
	2	.893	.932	.901	.109	.298
	3	.893	.932	.901	.109	.401

Table 10. Fraction correct classification for data in Fig. 3 with uniform noise with range from -10 to 10. See caption under Table 5.

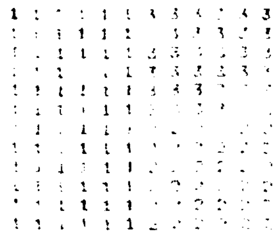
Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.722	.599	.760	.354	.493
Mean	2	.713	.599	.747	.353	.421
	3	.790	.675	.886	.301	.438
5 by 5	1	.865	.835	.899	.192	.380
Mean	2	.865	.835	.899	.192	.339
	3	.873	.832	.912	.148	.375
7 by 7	1	.837	.812	.841	.122	.315
Mean	2	.862	.847	.864	.106	.270
	3	.862	.847	.864	.106	.358
MM	1	.854	.813	.868	.219	.311
	2	.787	.721	.828	.216	.391
	3	.858	.780	.905	.185	.371
MMM	1	.866	.835	.866	.179	.304
	2	.829	.822	.818	.229	.348
	3	.822	.760	.841	.177	.324

Table 11. Fraction correct classification for data in Fig. 1 with uniform noise with range from -15 to 15. See caption under Table 5.

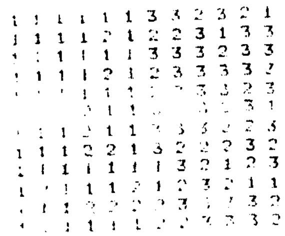
Filter	Method	Fraction Correct Output	Clean1	Clean2	Dispersion	
3 by 3	1	.783	.741	.857	.252	.517
Mean	2	.747	.660	.814	.328	.539
	3	.753	.686	.842	.330	.471
5 by 5	1	.833	.769	.861	.165	.375
Mean	2	.823	.774	.851	.165	.375
	3	.793	.719	.818	.149	.458
7 by 7	1	.812	.822	.816	.244	.266
Mean	2	.816	.785	.835	.158	.293
	3	.802	.785	.822	.158	.293
MM	1	.816	.760	.852	.197	.420
	2	.816	.760	.852	.197	.413
	3	.816	.760	.852	.197	.413
MMM	1	.835	.789	.853	.195	.355
	2	.831	.785	.847	.195	.361
	3	.841	.789	.857	.162	.414

Table 12. Fraction correct classification for data in Fig. 3 with uniform noise with range from -20 to 20. See caption under Table 5.

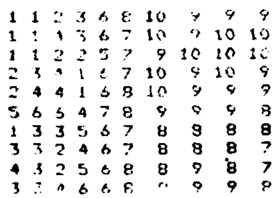
24



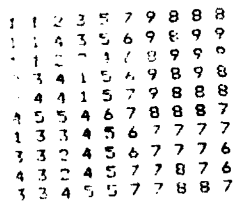
(a) Original clusters



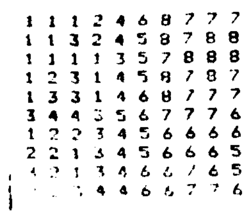
(b) Cluster means 40,55 and 60.
Noise gaussian with variance 64.
Cluster to nearest of 40, 55 and 60.



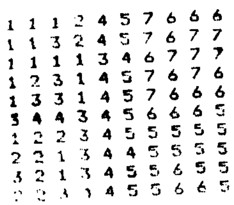
(c) Iteration 1.
D = 1.000



(d) Iteration 2.
D = 0.875



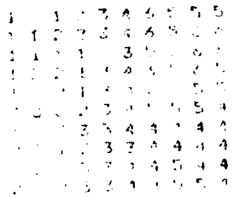
(e) Iteration 3.
D = 0.8333



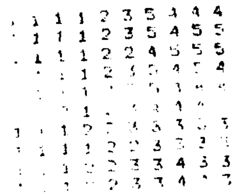
(f) Iteration 4.
D = 0.7778

Fig. 4 Example illustrating Method 1.

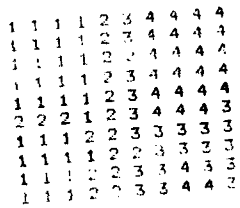
25



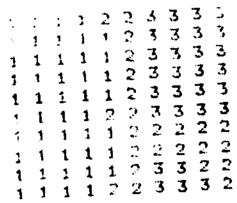
(g) Iteration 5.
D = 0.6250



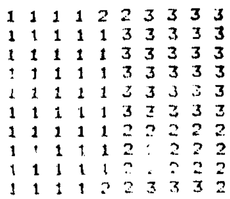
(h) Iteration 6.
D = 0.6071



(i) Iteration 7.
D = 0.5893



(j) Final iteration.
D = 0.4118



(k) Cleaned output.
See text for explanation.

Fig. 4 (continued).

Region	Noise range	Output	Clean1	Clean2	Dispersion	
1	-10 to 10	.6879	.8920	.8565	.5010	.6266
2	-20 to 20					
3	-40 to 40					
1	-5 to 5	.6953	.9127	.8698	.5090	.6316
2	-10 to 10					
3	-20 to 20					
1	-3 to 3	.7175	.7574	.7544	.5317	.63731
2	-20 to 20					
3	-30 to 30					
1	-2 to 2	.7526	.8402	.8299	.4283	.5815
2	-15 to 15					
3	-30 to 30					

Table 13. Fractions correct classification for data in Fig. 3 with uniform noise as indicated and expected value of 0 in each region. A gradient measure was used to distinguish the regions. See caption under Table 5, and text for further explanations.

5. Here the techniques were applied to a MCIDAS infrared temperature tape that was supplied by Robert Myers of the Air Force Geophysical Laboratory at Hanscom Air Force Base. The tape represents data in the Atlantic Ocean on 5 May, 1980. The first region we chose to analyze shows the Chesapeake Bay-Delaware Bay area down to Cape Hatteras as well as the edge of the Gulf Stream at the bottom right. The cluster method used was Method 1, and the prefilters were 3 by 3 Mean, 5 by 5 Mean and MM. Fig. 5 shows the output using the 5 by 5 Mean, Fig. 6 with the MM filter, and Fig. 7 the 3 by 3 Mean. Fig. 8 shows the result of taking the means of the regions produced by the MM filter, and clustering the original data to the nearest of these means. In each run THRSH was set at 0.23. The actual highest value of the dispersion for the resulting regions was 0.1982 for filter MM, 0.2191 for the 5 by 5 Mean, and 0.2258 for the 3 by 3 Mean In each case 6 clusters were produced. Note, however, the lack of resolution of the 5 by 5 Mean filter as compared to filter MM. Here were the means and standard deviations of the regions that were produced by these filters:

	5 by 5 Mean		MM		3 by 3 Mean	
Region	Mean	SD	Mean	SD	Mean	SD
1	61.9	4.93	60.2	2.55	59.0	2.16
2	70.8	4.90	67.2	3.95	65.2	3.20
3	78.8	3.72	78.0	5.14	76.7	5.41
4	87.2	2.30	87.1	2.15	87.5	2.05
5	90.0	0.98	90.0	1.12	90.5	0.95
6	93.2	1.32	93.2	1.32	93.2	1.41

To get some intuition for what these values represent, it should be noted that grey level values of 60, 67, 78, 87, 90 and 93 represent respectively Fahrenheit temperatures of 81.5, 75.2, 65.3, 57.2, 54.5 and 51.8 degrees. The next series of pictures represents the region obtained by shifting the view 16 rows down and 53 columns to the right. The two warm regions in

the upper left corner represent land masses bordering the Delaware Bay, and the cold region at the bottom of the picture is a cloud. The Gulf Stream lies immediatedly above and is partially obscured by this cloud. Fig. 9 represents the output of a 5 by 5 Mean filter, Fig. 10 is the MM filter, Fig. 11 the MMM filter, and Fig. 12 a 3 by 3 Mean. The statistics for these regions are:

Region	5 by 5 Mean		MM		MMM		3 by 3 Mean	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	73.32	2.33	73.56	2.52	74.5	4.36	73.25	1.13
2	78.39	3.59	78.74	3.74	79.0	4.30	78.52	3.16
3	86.01	3.52	87.32	2.86	87.54	4.52	87.14	2.29
4	89.81	2.18	90.16	2.20	90.57	3.61	90.25	1.17
5	103.2	23.85	103.8	23.80	100.9	23.66	106.7	24.18

The reason for the high SD in Region 5 is that it consists of a region of cold water as well as a cloud region. As a final note, it is important to note that no postcleaning oeration was performed on any of this data. The pictures represent the raw output of the cluster method. Also, the only parameter that was supplied was THRSH, and this was left set at 0.23. Based on a study made by Gerson et al [5], it is anticipated that a much better picture will result when more than a single feature is used as a basis clustering.

6. Directions for the future. One of the difficulties with all of the techniques of the paper is their dependence on the value of the parameter THRSH. The problem is caused by the fact that both noise and boundary points tend to increase the value of the dispersion of a region. Thus a region with a relatively long boundary will tend to have a higher dispersion than a region with a short boundary. An attempt will be made to either modify the dispersion measure so as to take this into account, or else supplement the dispersion measure by some other measure

that will ignore boundary points and concentrate on sorting out the noise.

A second project will be to see if the methods can be made more efficient when more than a single feature is considered. Finally, the techniques will be applied to standard data sets so that their ouput can be compared with that provided by other image segmentation algorithms.

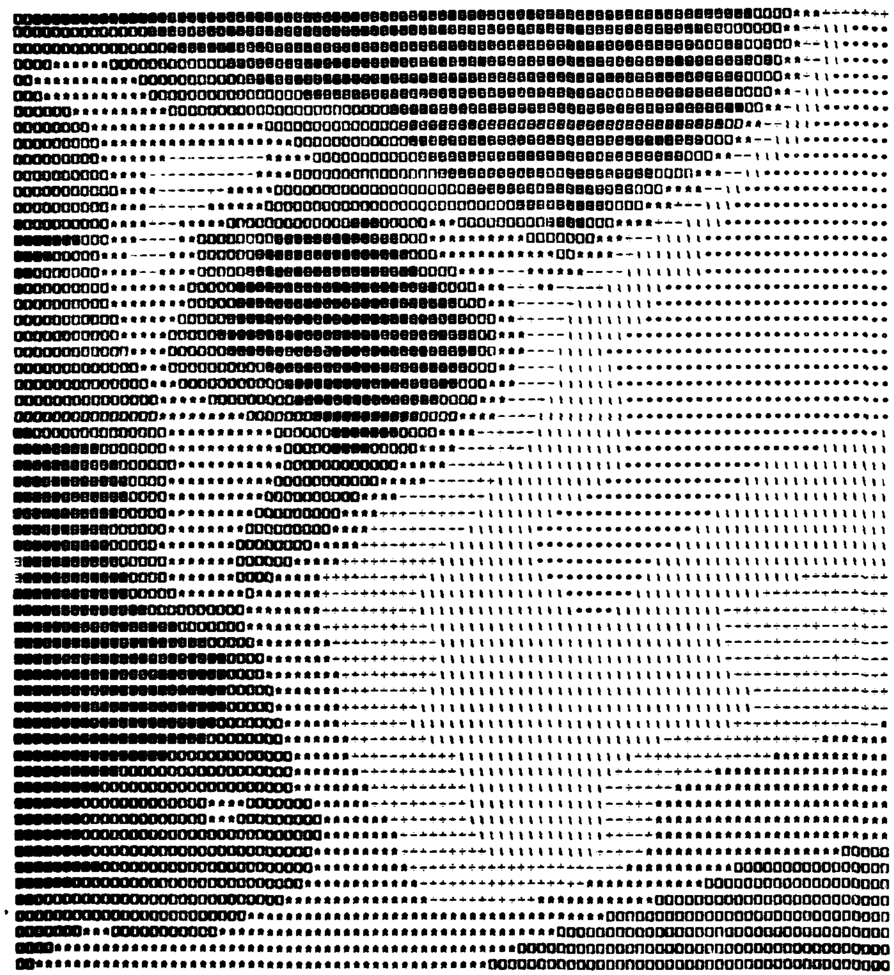


Fig. 5 Tempterature data using 5 by 5 Mean Filter.

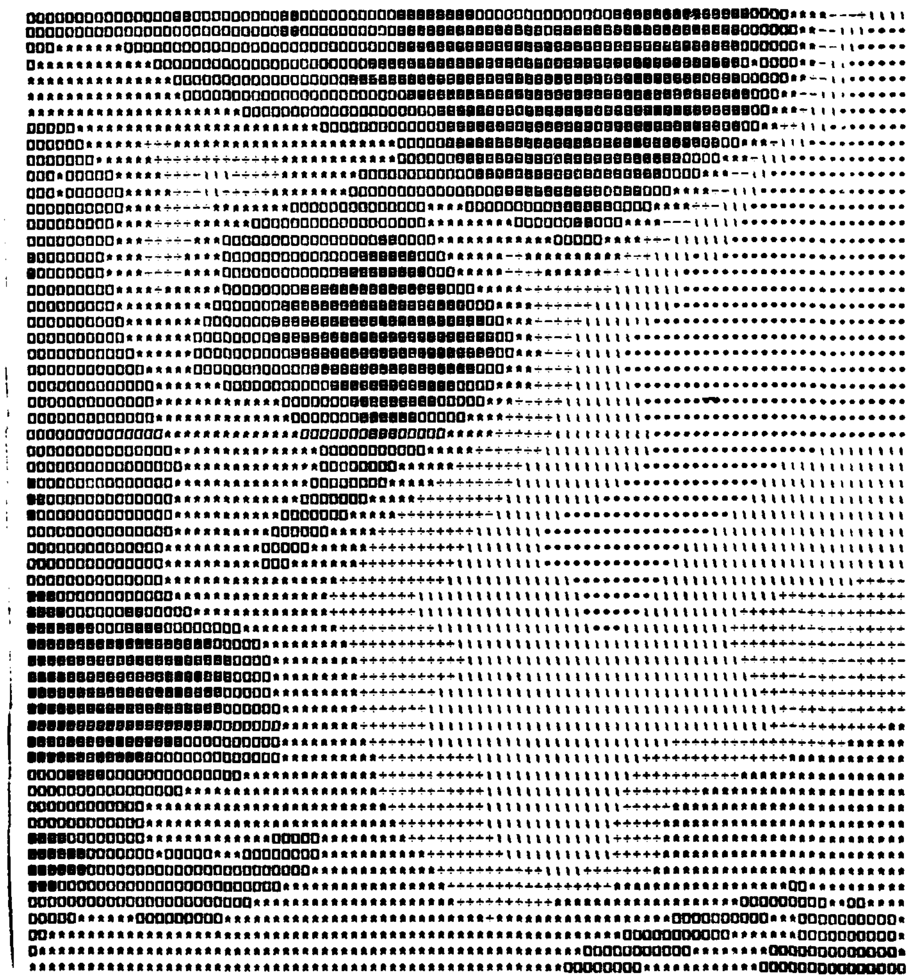


Fig. 6 Tempterature data using MM filter.

Fig. 7. Tempterature data using 3 by 3 Mean filter.

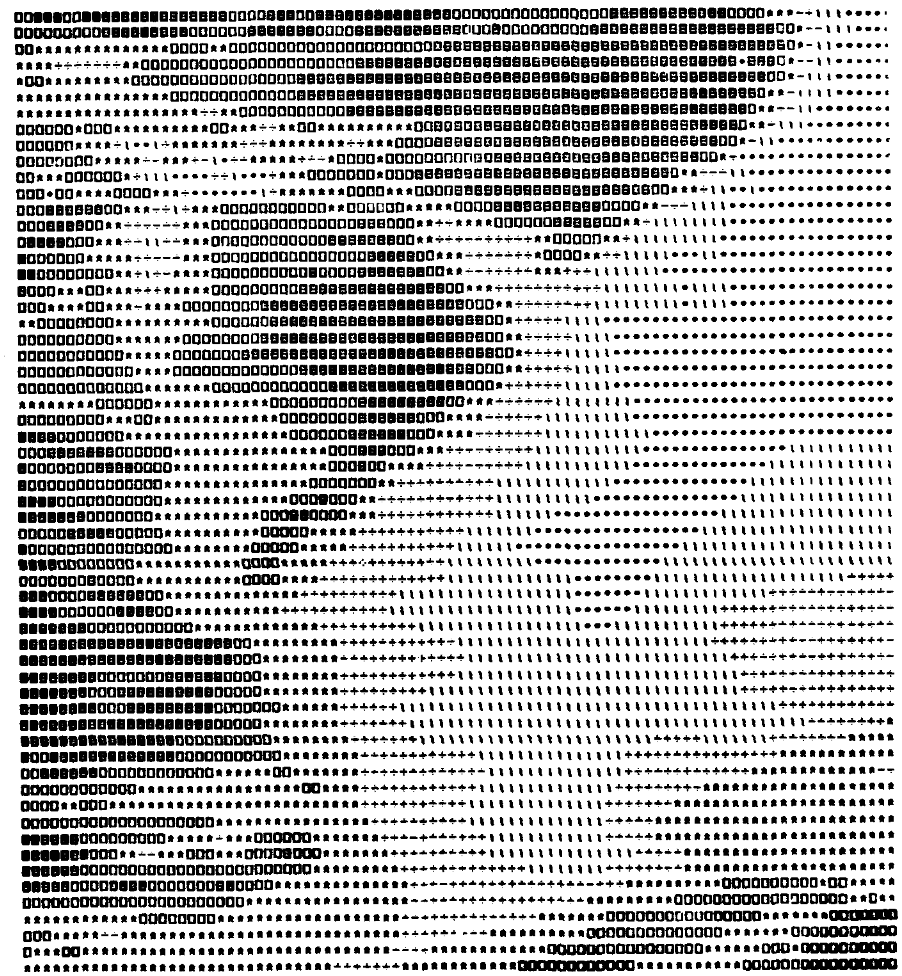


Fig. 8 Original temperature data clustered to nearest of means of regions found by MM filter.

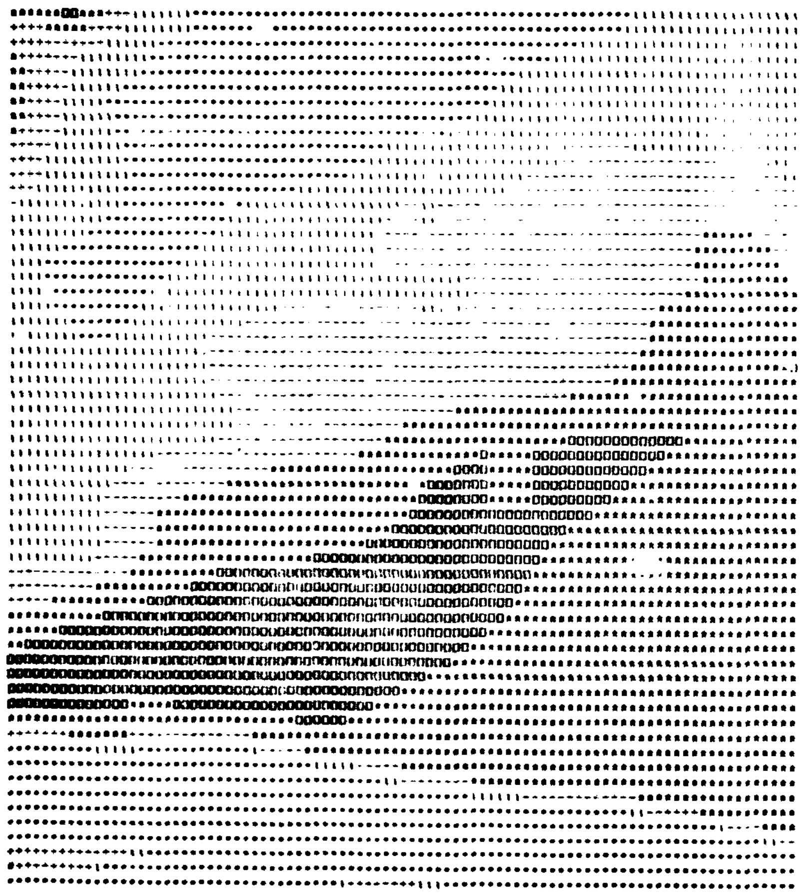


Fig. 9 Temperature data using 5 by 5 Mean Filter

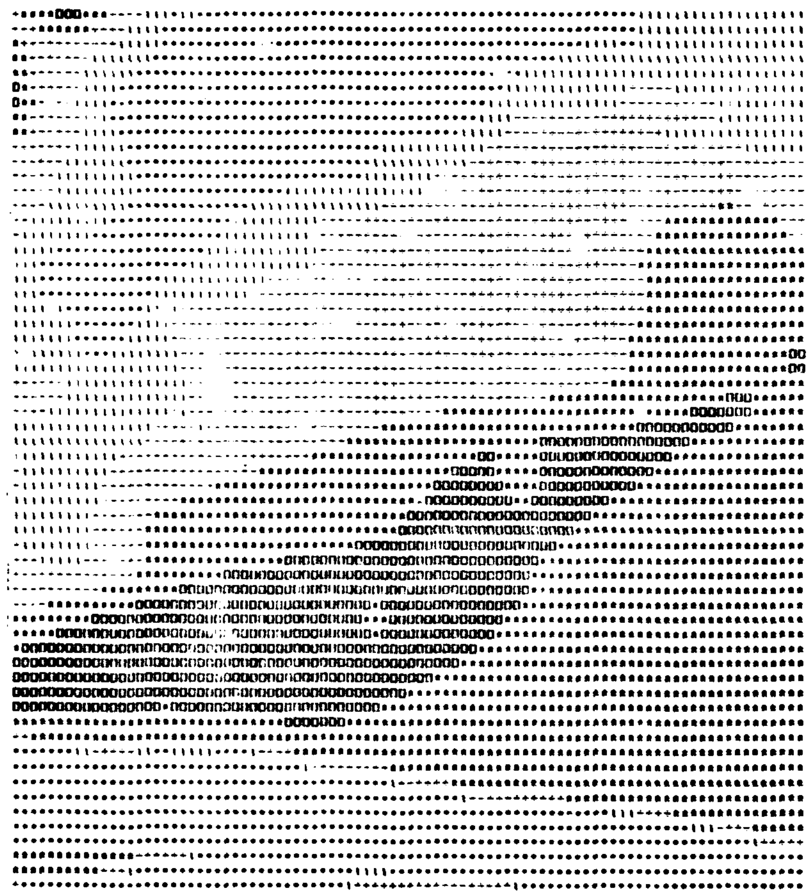


Fig. 10 Temperature data using MM filter

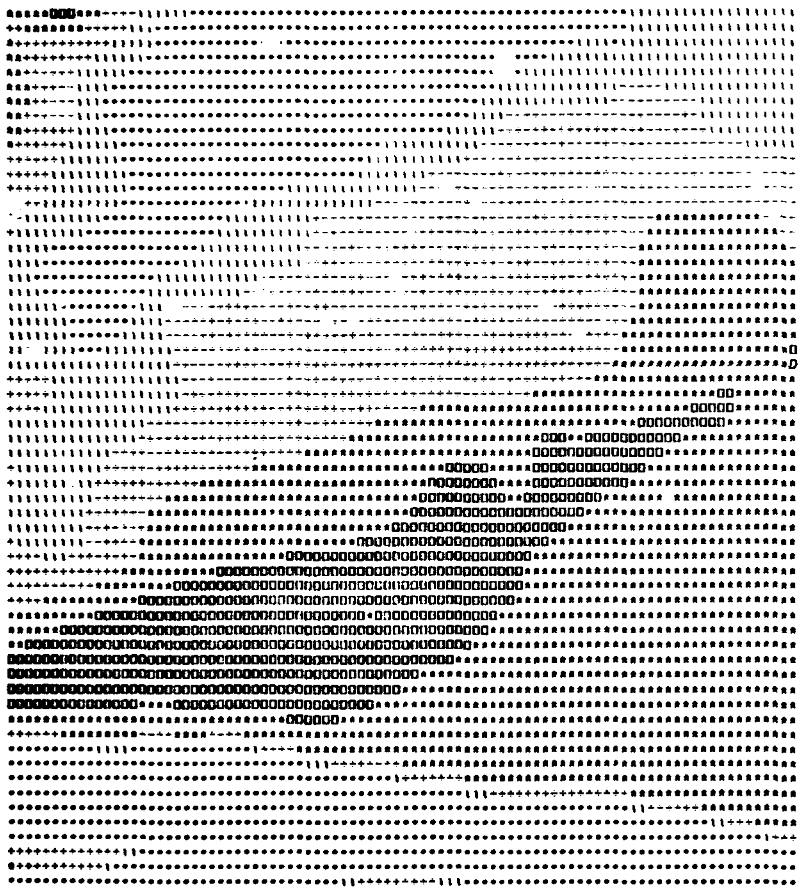


Fig. 11 Temperature data using [illegible] filter

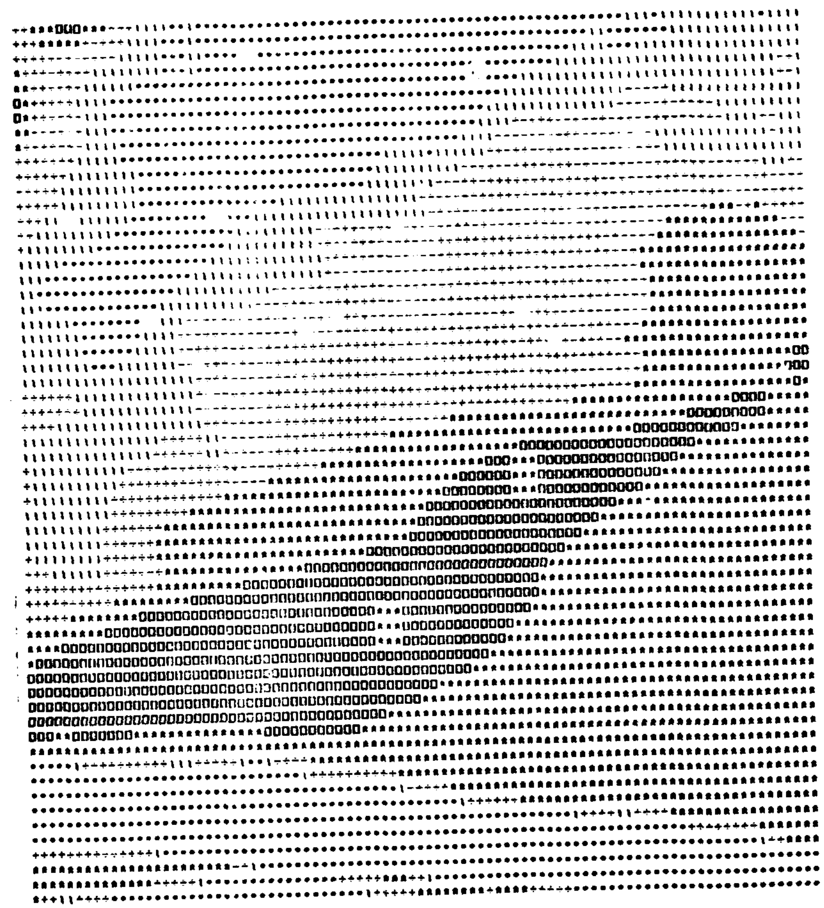


Fig. 12 Temperature data using 3 by 3 Mean Filter

REFERENCES

[1] Bryant, Jack, On the clustering of multidimensional pictorial data, Pattern Recognition 11 (1979) 115-125.

[2] Coleman, Guy, B and Andrews, Harry C., Image segmentation by clustering, Proc. IEEE 67 (1979) 773-785.

[3] Riordan, John, An introduction to combinatorial analysis, Wiley (1958), New York, 244 pp.

[4] Rosenfeld, A. and Kak, A., Digital Picture Processing, Academic Press (1976) New York, 457 pp.

[5] Gerson, D., Khedouri, E. and Gaborski, P., Locating ocean fronts using digital satellite temperature data and AUTOMATED PATTERN ANALYSIS, Gulf Stream 5 (1979), 3-7.

Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA 01003

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: TR-J8102 (14)
2. GOVT ACCESSION NO.: AD-A107 110
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Cluster Analysis Algorithms For Image Segmentation. (6)
5. TYPE OF REPORT & PERIOD COVERED: Technical rept. (9)
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): M. P. Janowitz (10)
8. CONTRACT OR GRANT NUMBER(s): N00014-79-C-0629 (15)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Massachusetts, Amherst, MA 01003
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 121405
11. CONTROLLING OFFICE NAME AND ADDRESS: Procuring Contract Officer, Office of Naval Research, Arlington, VA 22217
12. REPORT DATE: Jul 1981 (11)
13. NUMBER OF PAGES: 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research Resident Representative, Harvard University, Gordon McKay Laboratory, Room 113, Cambridge, MA 02138
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED. (12) 34
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Image segmentation, Cluster analysis, Pattern analysis
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

Several clustering algorithms for image segmentation are described and compared. They are applied to both simulated data and to MCIDAS ocean temperature data.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

81 10 29 090

DATE
ILMED
8